Cent. Eur. J. Math. • 1-18
Author version

Central European Journal of Mathematics

A parameter-free stabilized finite element method for scalar advection-diffusion problems

Research

Pavel Bochev[13][*], **Kara Peterson**[23][†]

1 Numerical Analysis and Applications,
Sandia National Laboratories, Mail Stop 1320
Albuquerque, New Mexico, 87185-1320

2 Numerical Analysis and Applications,
Sandia National Laboratories, Mail Stop 1320
Albuquerque, New Mexico, 87185-1320

3 Sandia National Laboratories is a multi-program laboratory operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Abstract: We formulate and study numerically a new, parameter-free stabilized finite element method for advection-diffusion problems. Using properties of compatible finite element spaces we establish connection between nodal diffusive fluxes and one-dimensional diffusion equations on the edges of the mesh. To define the stabilized method we extend this relationship to the advection-diffusion case by solving simplified one-dimensional versions of the governing equations on the edges. Then we use $H(curl)$-conforming edge elements to expand the resulting edge fluxes into an exponentially fitted flux field inside each element. Substitution of the nodal flux by this new flux completes the formulation of the method. Utilization of edge elements to define the numerical flux and the lack of stabilization parameters differentiate our approach from other stabilized methods. Numerical studies with advection-diffusion test problems on a range of structured and unstructured grids confirm the excellent stability and robustness of the new method.

MSC: AMS Mathematics Subject Classification (2000) numbers: 65N30, 65N12

Keywords: Advection-diffusion • upwind stabilization • exponentially fitted flux • finite element method • edge elements

[*] *E-mail: pbboche@sandia.gov*
[†] *E-mail: kjpeter@sandia.gov*

I. Introduction

In this paper we formulate and study a new, parameter-free stabilized finite element method for the scalar advection-diffusion equation

$$\begin{cases} -\nabla\cdot(\varepsilon\nabla\phi-\mathbf{u}\phi)=f & \text{in } \Omega \\ \phi=g & \text{on } \Gamma_D \\ \boldsymbol{n}\cdot(\varepsilon\nabla\phi-\mathbf{u}\phi)=h & \text{on } \Gamma_N\,. \end{cases} \tag{1}$$

Our main focus is on advection-dominated problems where the diffusion coefficient ε is much smaller than the advective velocity vector $\mathbf{u}$. It is well-known that in this case Galerkin methods for (1) can develop spurious oscillations if the mesh size does not resolve solution features such as internal and/or boundary layers. Because for small ε the required level of mesh refinement is impractical, stabilized Galerkin methods [3, 5, 9, 11, 12] are a popular alternative. We refer to [7, 14, 15] for numerical studies of various stabilized methods.

Many stabilized Galerkin methods involve a mesh-dependent stabilization parameter. The quality of the finite element solution depends critically on the choice of this parameter. Yet, finding the optimal stabilization parameters for different settings remains an open problem. Dependence of this parameter on constants that are known exactly only in special cases [10] is one part of the puzzle. Another is that different solution features, such as interior layers and boundary layers along Dirichlet boundaries may require different selection strategies; see [14]. As a result, definition of stabilization parameters often relies on heuristic arguments that may depend on the mesh configuration and/or the solution features.

In this paper we formulate a parameter-free stabilized finite element method for (1) using the lowest-order nodal and edge elements. Using the fact that these finite element spaces belong to an exact sequence [2] we first establish connection between nodal Galerkin diffusive fluxes and one-dimensional diffusion equations on the edges of the mesh. To define our stabilized method we extend this relationship to the advection-diffusion case by solving simplified one-dimensional versions of the governing equations (1) on the edges. Then we use the edge elements to expand the resulting edge fluxes into an exponentially fitted flux field inside each element. Substitution of the nodal Galerkin flux by this new flux completes the formulation of the method. This substitution modifies the trial space and differentiates our method from other Petrov-Galerkin stabilized methods, which modify the test space. Utilization of edge elements to define the numerical flux and the lack of stabilization parameters are two other distinguishing characteristics of our method, which we call Exponentially Fitted Flux Petrov-Galerkin, or EFF-PG for short. One can view the EFF-PG as a finite element extension of the classical Scharfetter-Gummel (SG) idea [18] to unstructured grids.

Preliminary computational studies reveal that the EFF-PG method is flexible, robust, and performs reliably over a wide range of test problems and Peclet numbers without requiring user intervention for tuning or calibration. Our method is less dissipative than the classical artificial diffusion method, but more dissipative than the SUPG [11]. However, it handles both internal and boundary layers, and does not exhibit significant crosswind oscillations, which require the additional discontinuity capturing (DC) term [12] in the SUPG. We recall that this term depends on the solution itself, thereby turning SUPG+DC into a nonlinear method, even though (1) is linear.

However, the attractive computational properties of our method are not without some costs. The EFF-PG weak equation is not a weighted residual formulation, i.e., the method is not consistent in the usual Galerkin sense. Thus, using the lowest-order C^0 element yields a first-order accurate formulation. Higher-order elements can be an effective way to improve the accuracy of the method and are the subject of a forthcoming paper.

Exponentially fitted conforming finite elements [1, 17, 20, 21] are another approach with similar roots. These methods define exponentially fitted $H^1(\Omega)$-conforming basis functions using one-dimensional advection-diffusion problems similar to the ones we solve on the edges of the mesh. However, the exponentially fitted basis functions and their derivatives are not known in closed form. Computation of their point values requires solution of multiple one-dimensional problems [20]. Consequently, the cost of these methods is proportional to the number of integration points in the mesh. In contrast, in our method the number of one-dimensional problems is always equal to the number of edges in the mesh. In addition, extension of edge fluxes into exponentially fitted element fluxes uses only standard edge elements.

A related, yet different approach is the stabilized Generalized Finite Element method (GFEM) [19]. This method uses the partition of unity property of Lagrangian nodal shape functions to incorporate "enrichment" functions which capture the exponential character of the exact solution of (1) in the advection-dominated case. As with other GFEM, this method requires weak enforcement of Dirichlet boundary conditions by penalty. In addition to choosing the penalty parameter, the user must also choose the level of enrichment (enriched, partially enriched or unenriched) for the nodes in the mesh, i.e., this method continues to require a non-trivial level of user intervention. In contrast, our formulation does not require any such intervention and adjusts automatically to the salient problem features.

Our paper is organized as follows. The rest of this section introduces the basic notation. Section II presents the EFF-PG method. In Section II.3 we discuss implementation of the method and Section II.4 provides some information about the resulting algebraic equations and their interpretation. Section III contains numerical studies and Section IV summarizes our findings.

I.1. Notation

In this paper Ω is a bounded region in $\Re^n$, $n = 2, 3$ with Lipschitz-continuous boundary $\partial\Omega$. The Neumann and Dirichlet parts of the boundary are Γ_N and Γ_D, respectively, and $H^k(\Omega)$ is the Sobolev space of order k. When $k = 0$ we use the standard notation $L^2(\Omega)$. The space of all square integrable vector fields whose curl is also square integrable is $H(curl, \Omega)$.

Throughout the paper $K_h(\Omega)$ is a conforming finite element partition of Ω into elements K_s with size h_s. The average size of the elements in the mesh is h. The vertices of the mesh are $\boldsymbol{v}_i$ and $\boldsymbol{e}_{ij}$ is a mesh edge with endpoints $\boldsymbol{v}_i$ and $\boldsymbol{v}_j$. The midpoint and the length of $\boldsymbol{e}_{ij}$ are

$$\boldsymbol{m}_{ij} = \frac{\boldsymbol{v}_i + \boldsymbol{v}_j}{2} \quad \text{and} \quad h_{ij} = |\boldsymbol{v}_i - \boldsymbol{v}_j|\,,$$

respectively. The vertices and the edges intersecting with entity $*$ are $V(*)$ and $E(*)$, respectively. For example, $V(\Omega)$ is the set of all mesh vertices, $E(\Omega)$ is the set of all mesh edges, $V(K_s)$ are the vertices of element K_s, $E(\boldsymbol{v}_i)$ are all edges having $\boldsymbol{v}_i$ as a vertex, and so on.

Selection of a vertex ordering induces orientation of the edges in $E(\Omega)$. There are exactly two ways to order the vertices $\boldsymbol{v}_i$ and $\boldsymbol{v}_j$ of an edge $\boldsymbol{e}_{ij}$: either $\boldsymbol{v}_i$ is the first vertex or $\boldsymbol{v}_j$ is the first vertex. Because the edge notation is $\boldsymbol{e}_{ij}$ regardless of the vertex order, we encode the orientation of $\boldsymbol{e}_{ij}$ by an integer σ_{ij}:

$$\sigma_{ij} = \begin{cases} -1 & \text{if } \boldsymbol{v}_i \text{ is the first vertex of } \boldsymbol{e}_{ij}\text{, i.e., the vertex order is } \boldsymbol{v}_i \to \boldsymbol{v}_j \\ \;\;1 & \text{if } \boldsymbol{v}_i \text{ is the second vertex of } \boldsymbol{e}_{ij}\text{, i.e., the vertex order is } \boldsymbol{v}_i \leftarrow \boldsymbol{v}_j \end{cases} \tag{2}$$

The oriented unit tangent on $\boldsymbol{e}_{ij}$ always points towards the *second* vertex of the edge:

$$\boldsymbol{t}_{ij} = \sigma_{ij} \frac{\boldsymbol{v}_i - \boldsymbol{v}_j}{|\boldsymbol{v}_i - \boldsymbol{v}_j|}\,.$$

The main goal of this paper is to present the basic idea of our approach. For this reason we formulate the EFF-PG method using the lowest-order $H^1(\Omega)$-conforming and $H(curl, \Omega)$-conforming finite element spaces, which provide a simple, yet sufficiently complete setting for the method. We denote these spaces by $\mathbf{G}^h(\Omega)$ and $\mathbf{C}^h(\Omega)$, respectively. Thus, $\mathbf{G}^h(\Omega)$ is the C^0 piecewise linear, bilinear or trilinear finite element space and $\mathbf{C}^h(\Omega)$ is the lowest-order Nedelec edge element space [16]. The latter contains piecewise smooth vector fields whose tangential component is continuous along the element edges, thus the monicker "edge elements". The basis of $\mathbf{G}^h(\Omega)$ is $\{N_i\}$, $\boldsymbol{v}_i \in V(\Omega)$. We assume that N_i is the standard nodal, or Lagrangian basis:

$$N_i(\boldsymbol{v}_j) = \delta_i^j\,. \tag{3}$$

The basis of $\mathbf{C}^h(\Omega)$ is $\{\vec{W}_{ij}\}$, $\boldsymbol{e}_{ij} \in E(\Omega)$. There are two standard unisolvent sets of degrees of freedom[1] for the lowest-order Nedelec element: the mean of the tangent component of $\vec{W}_{ij}$ along an edge, or the value of this component at the edge midpoint. In this paper we use the former, that is, basis functions have the property

$$\int_{\boldsymbol{e}_{kl}} \vec{W}_{ij} \cdot \boldsymbol{t}_{kl} d\ell = \delta_{ij}^{kl}\,. \tag{4}$$

With this choice at the edge midpoint

$$\vec{W}_{ij} \cdot \boldsymbol{t}_{kl}\Big|_{\boldsymbol{m}_{kl}} = \frac{\delta_{ij}^{kl}}{h_{kl}}\,. \tag{5}$$

[1] *The basis functions corresponding to the two unisolvent sets differ only by a scaling factor.*

We note that $\vec{W}_{ij} \cdot \boldsymbol{t}_{ij} > 0$, i.e., orientation of the edge basis function $\vec{W}_{ij}$ always follows the orientation of the associated edge $\boldsymbol{e}_{ij}$.

When Γ_D is non-empty we also need the subspace $\mathbf{G}^h_D(\Omega)$ of all functions in $\mathbf{G}^h(\Omega)$, which vanish on Γ_D, and the subspace $\mathbf{C}^h_D(\Omega)$ of all fields in $\mathbf{C}^h(\Omega)$ whose tangential component vanishes on Γ_D.

II. Formulation of the method

The standard Galerkin finite element method for (1) seeks a function

$$\phi_h = \sum_{\boldsymbol{v}_j \in V(\Omega)\setminus V(\Gamma_D)} n_j N_j + \sum_{\boldsymbol{v}_j \in V(\Gamma_D)} g(\boldsymbol{v}_j) N_j \in \mathbf{G}^h(\Omega) \tag{6}$$

such that

$$\int_\Omega F(\phi_h) \cdot \nabla \psi_h \, dV = \int_\Omega f \psi_h dV + \int_{\Gamma_N} h \psi_h dS \quad \forall \psi_h \in \mathbf{G}^h_D(\Omega)\,, \tag{7}$$

where $F(\phi_h) = (\varepsilon \nabla \phi_h - \mathbf{u}\phi_h)$. The vector field $F(\phi_h)$ is a nodal Galerkin approximation of the exact total flux $F(\phi) = (\varepsilon \nabla \phi - \mathbf{u}\phi)$. To motivate our approach we examine (7) in the pure diffusion limit and establish a relationship between the nodal Galerkin flux $F(\phi_h)$ and one-dimensional diffusion equations on the edges of the mesh.

Theorem II.1.

Assume that $\mathbf{u} = 0$ and $0 \le s \le h_{ij}$ is the natural length parameter on edge $\mathbf{e}_{ij}$. The nodal Galerkin flux admits the representation

$$F(\phi_h) = \sum_{\boldsymbol{e}_{ij} \in E(\Omega)} F^0_{ij} \vec{W}_{ij}\,, \tag{8}$$

where the edge fluxes

$$F^0_{ij} = h_{ij} \varepsilon \frac{d\phi(s)}{ds} \qquad \forall \boldsymbol{e}_{ij} \in E(\Omega) \tag{9}$$

and $\phi(s)$ solves the one-dimensional edge diffusion equation

$$\begin{cases} -\varepsilon \dfrac{d^2\phi(s)}{ds^2} = 0 \quad \text{on } \boldsymbol{e}_{ij} \\ \phi(0) = n_i \quad \text{and} \quad \phi(h_{ij}) = n_j\,. \end{cases} \tag{10}$$

The nodal values of ϕ_h provide the boundary data in (10).

Proof. For clarity[2] we present the proof for pure Neumann conditions when (6) reduces to

$$\phi_h = \sum_{\boldsymbol{v}_i \in V(\Omega)} n_i N_i\,. \tag{11}$$

[2] *The case $\Gamma_D \neq \emptyset$ involves edges with two different kind of vertices: one in the interior where ϕ_h is unknown and one on Γ_D where ϕ_h is given. Handling these cases complicates notation without bringing additional insights.*

With this assumption, in the pure diffusion limit

$$F(\phi_h) = \varepsilon \nabla \phi_h = \sum_{\boldsymbol{v}_i \in V(\Omega)} \varepsilon n_i \nabla N_i \,. \tag{12}$$

The exact sequence property of compatible finite element spaces [2] implies that $\nabla \mathbf{G}^h(\Omega) \subset \mathbf{C}^h(\Omega)$, and in particular, $\nabla N_i \in \mathbf{C}^h(\Omega)$. Moreover, for the lowest order nodal and edge finite element spaces there holds [4]

$$\nabla N_i = \sum_{\boldsymbol{e}_{ij} \in E(\boldsymbol{v}_i)} \sigma_{ij} \vec{W}_{ij} \,. \tag{13}$$

Combining (12) and (13) yields

$$F(\phi_h) = \sum_{\boldsymbol{v}_i \in V(\Omega)} \varepsilon n_i \left(\sum_{\boldsymbol{e}_{ij} \in E(\boldsymbol{v}_i)} \sigma_{ij} \vec{W}_{ij} \right) \,. \tag{14}$$

Without loss of generality we may assume that the vertices of $\boldsymbol{e}_{ij}$ are ordered as $\boldsymbol{v}_i \to \boldsymbol{v}_j$ and so, $\sigma_{ij} = -1$. After exchanging the order of summation in (14) we arrive at

$$F(\phi_h) = \sum_{\boldsymbol{e}_{ij} \in E(\Omega)} \varepsilon (n_j - n_i) \vec{W}_{ij} \tag{15}$$

On the other hand, a straightforward calculation shows that

$$\phi(s) = n_i + s \frac{n_j - n_i}{h_{ij}}$$

is the exact solution of (10). As a result,

$$\varepsilon (n_j - n_i) = h_{ij} \left(\varepsilon \frac{d\phi(s)}{ds} \right) = F^0_{ij} \,,$$

which proves the theorem. □

Returning to the general case, recall that the nodal flux $F(\phi_h) = (\varepsilon \nabla \phi_h - \mathbf{u} \phi_h)$ is not appropriate for advection-dominated problems and solutions of (7) can develop spurious oscillations when $\varepsilon \ll |\mathbf{u}|$. The representation (8) prompts us to seek an alternative to $F(\phi_h)$ in terms of edge elements

$$F(\phi) \approx F_h(\phi_h) = \sum_{\boldsymbol{e}_{ij} \in E(\Omega)} F_{ij} \vec{W}_{ij} \,, \tag{16}$$

where F_{ij} are edge fluxes that incorporate the local behavior of the exact solution of (1). Theorem II.1 reveals that in the diffusion limit these fluxes correspond to pure diffusion problems on each edge. Accordingly, in the general case we propose to define F_{ij} by solving one-dimensional advection-diffusion equations on the mesh edges.

II.1. The discrete edge fluxes.

We retain the assumption that $\boldsymbol{e}_{ij}$ has orientation $\sigma_{ij} = -1$. Along $\boldsymbol{e}_{ij}$ we consider the following one-dimensional boundary value problem (BVP)

$$\begin{cases} -\dfrac{d}{ds}\left(\varepsilon_{ij}\dfrac{d\phi(s)}{ds} - u_{ij}\phi(s)\right) = 0 \quad \text{for } 0 < s < h_{ij} \\ \phi(0) = n_i \quad \text{and} \quad \phi(h_{ij}) = n_j \end{cases} \tag{17}$$

where

$$u_{ij} = \mathbf{u}\cdot\boldsymbol{t}_{ij} \quad \text{and} \quad \varepsilon_{ij} = \varepsilon|_{\boldsymbol{e}_{ij}}\,.$$

are the edge velocity and diffusion, respectively. This BVP extends (10) to the general advection-diffusion case. As before, the unknown nodal coefficients n_i and n_j of the finite element solution (6) specify the boundary data. To solve (17) we approximate u_{ij} and ε_{ij} by their mean edge values $\bar{u}_{ij}$ and $\bar{\varepsilon}_{ij}$, respectively. A straightforward calculation reveals that the exact solution of the simplified edge equation is

$$\phi(s) = \frac{\exp(\alpha_{ij})n_i - n_j}{\exp(\alpha_{ij}) - 1} + \frac{n_j - n_i}{\exp(\alpha_{ij}) - 1}\exp(\bar{u}_{ij}s/\bar{\varepsilon}_{ij}) \tag{18}$$

where

$$\alpha_{ij} = \frac{\bar{u}_{ij}h_{ij}}{\bar{\varepsilon}_{ij}}$$

is the edge Peclet number. Following (10) we define the edge fluxes using the total flux of (18):

$$F_{ij} = h_{ij}\left(\bar{\varepsilon}_{ij}\frac{d\phi(s)}{ds} - \bar{u}_{ij}\phi(s)\right) = h_{ij}\bar{u}_{ij}\frac{n_j - \exp(\alpha_{ij})n_i}{\exp(\alpha_{ij}) - 1}\,. \tag{19}$$

To develop a more computationally stable expression we multiply and divide this formula by $\exp(-\beta_{ij})$, where $\beta_{ij} = \alpha_{ij}/2$. This yields

$$F_{ij} = h_{ij}\bar{u}_{ij}\frac{\exp(-\beta_{ij})n_j - \exp(\beta_{ij})n_i}{\exp(\beta_{ij}) - \exp(-\beta_{ij})}\,.$$

Few simple steps transform this formula into

$$F_{ij} = \frac{h_{ij}\bar{u}_{ij}}{2}\Big[n_j(\coth(\beta_{ij}) - 1) - n_i(\coth(\beta_{ij}) + 1)\Big]\,. \tag{20}$$

Remark II.1.

In (19) the scaling of the total flux $\bar{\varepsilon}_{ij}\phi'(s) - \bar{u}_{ij}\phi(s)$ by h_{ij} is dictated by the choice of the degrees of freedom for the edge element basis $\{\vec{W}_{ij}\}$. Indeed, since $F_h(\phi_h)$ approximates $F(\phi)$, property (5) implies that

$$F(\phi)\cdot\boldsymbol{t}_{kl}\Big|_{\boldsymbol{m}_{kl}} \approx F_h(\phi_h)\cdot\boldsymbol{t}_{kl}\Big|_{\boldsymbol{m}_{kl}} = \sum_{\boldsymbol{e}_{ij}\in E(\Omega)} F_{ij}\left(\vec{W}_{ij}\cdot\boldsymbol{t}_{kl}\right)\Big|_{\boldsymbol{m}_{kl}} = \frac{F_{kl}}{h_{kl}}\,,$$

and so,

$$F_{kl} \approx h_{kl}F(\phi)\cdot\boldsymbol{t}_{kl}\Big|_{\boldsymbol{m}_{kl}}\,. \tag{21}$$

Changing the degrees of freedom from mean edge values to midpoint edge values changes the scaling of the basis functions. Instead of (5) we have

$$\vec{W}_{ij} \cdot \boldsymbol{t}_{kl}\Big|_{\boldsymbol{m}_{kl}} = \delta_{ij}^{kl}\,, \tag{22}$$

and, instead of (21) - the identity

$$F_{kl} \approx F(\phi) \cdot \boldsymbol{t}_{kl}\Big|_{\boldsymbol{m}_{kl}}\,,$$

In this case the discrete edge flux (19) does not require the edge length factor h_{ij}.

II.2. The stabilized formulation.

Using the edge fluxes (20) as coefficients in (16) yields

$$F_h(\phi_h) = \sum_{\boldsymbol{e}_{ij} \in E(\Omega)} F_{ij}\vec{W}_{ij} = \sum_{\boldsymbol{e}_{ij} \in E(\Omega)} \frac{h_{ij}\bar{u}_{ij}}{2}\Big[n_j(\coth(\beta_{ij}) - 1) - n_i(\coth(\beta_{ij}) + 1)\Big]\vec{W}_{ij}\,. \tag{23}$$

The vector field (23) belongs in $\mathbf{C}^h(\Omega)$ and defines an exponentially fitted flux on Ω. To complete the formulation of the EFF-PG method we replace the nodal Galerkin flux $F(\phi_h)$ in (7) by $F_h(\phi_h)$. Therefore, the new method seeks $\phi_h \in \mathbf{G}^h(\Omega)$ such that

$$\int_\Omega F_h(\phi_h) \cdot \nabla\psi_h\, dV = \int_\Omega f\psi_h dV + \int_{\Gamma_N} h\psi_h dS \quad \forall \psi_h \in \mathbf{G}_D^h(\Omega)\,. \tag{24}$$

The use of (23) stabilizes the variational equation (24) without an explicit stabilization parameter. As a result, the EFF-PG method does not require additional tuning or calibration to the problem on hand.

While the EFF-PG is not a weighted residual formulation, in the pure diffusion limit it recovers the standard Galerkin method for the Poisson equation. The following lemma establishes this fact.

Lemma II.1.

Assume that ε is constant and (23) defines $F_h(\phi_h)$. Then

$$\lim_{\mathbf{u}\to 0} \int_\Omega F_h(\phi_h) \cdot \nabla\psi_h\, dV = \int_\Omega \varepsilon\nabla\phi_h \cdot \nabla\psi_h\, dV\,. \tag{25}$$

Proof. It is straightforward to check that

$$\lim_{\mathbf{u}\to 0} \bar{u}_{ij}\coth(\beta_{ij}) = \frac{2\varepsilon}{h_{ij}}\,.$$

As a result,

$$\lim_{\mathbf{u}\to 0} F_{ij} = \frac{h_{ij}\bar{u}_{ij}}{2}\Big[n_j(\coth(\beta_{ij}) - 1) - n_i(\coth(\beta_{ij}) + 1)\Big]\vec{W}_{ij} = \varepsilon(n_j - n_i) = F_{ij}^0\,.$$

Therefore, in the diffusion limit $F_h(\phi_h)$ converges to the nodal Galerkin flux $F(\phi_h) = \varepsilon\nabla\phi_h$. This proves the lemma. □

II.3. Implementation

To discus implementation of the EFF-PG note that the right hand side in (24) has the form

$$\sum_{\boldsymbol{e}_{kl}\in E(\Omega)} \frac{h_{kl}\bar{u}_{kl}}{2}\Big[n_l(\coth(\beta_{kl})-1)-n_k(\coth(\beta_{kl})+1)\Big]\,,\int_\Omega \vec{W}_{kl}\cdot\nabla\psi_h\,dV \tag{26}$$

and defines the discretization matrix $\mathbf{K}$ of the method. Formally, assembly of (26) requires loop over element edges, yet the unknowns are the nodal degrees of freedom of ϕ_h, i.e., the same as in a standard Galerkin or a stabilized method for (1). To facilitate reuse of existing code infrastructures for nodal finite elements it is desirable to provide an alternative vertex-based formulation of (26).

To this end we collect all terms in (23) that share the same nodal degree of freedom n_j associated with vertex $\boldsymbol{v}_j$. These terms correspond to the edges in $E(\boldsymbol{v}_j)$ and so, we arrive at an equivalent vertex-based expression for (23):

$$F_h(\phi_h)=\sum_{\boldsymbol{v}_j\in V(\Omega)}\sum_{\boldsymbol{e}_{jk}\in E(\boldsymbol{v}_j)}\sigma_{jk}\frac{h_{jk}\bar{u}_{jk}}{2}\Big[n_j(\coth(\beta_{jk})-\sigma_{jk})\Big]\vec{W}_{jk}\,. \tag{27}$$

The corresponding vertex-based formulation of the EFF-PG method is: seek $\phi_h\in\mathbf{G}^h(\Omega)$ such that

$$\sum_{\boldsymbol{v}_j\in V(\Omega)}\sum_{\boldsymbol{e}_{jk}\in E(\boldsymbol{v}_j)}\sigma_{jk}\frac{h_{jk}\bar{u}_{jk}}{2}\Big[n_j(\coth(\beta_{jk})-\sigma_{jk})\Big]\int_\Omega\vec{W}_{jk}\cdot\nabla\psi_h\,dV=\int_\Omega f\psi_h dV+\int_{\Gamma_N}h\psi_h dS\quad\forall\psi_h\in\mathbf{G}^h_D(\Omega)\,, \tag{28}$$

The weak problem (28) is equivalent to a linear algebraic system of equations

$$\mathbf{K}\boldsymbol{n}=\boldsymbol{f} \tag{29}$$

for the unknown coefficient vector $\boldsymbol{n}=(n_1,\dots,n_k)$ of ϕ_h. The matrix of this system has element

$$\mathbf{K}_{ij}=\sum_{\boldsymbol{e}_{jk}\in E(\boldsymbol{v}_j)}\sigma_{jk}\frac{h_{jk}\bar{u}_{jk}}{2}\big(\coth(\beta_{jk})-\sigma_{jk}\big)\int_\Omega\vec{W}_{jk}\cdot\nabla N_i\quad\forall\boldsymbol{v}_i,\boldsymbol{v}_j\in V(\Omega)\backslash V(\Gamma_D)\,. \tag{30}$$

Element $\mathbf{K}_{ij}$ involves products of edge basis functions and gradients of nodal basis functions. However, assembly of $\mathbf{K}_{ij}$ is possible through a conventional loop structure for standard Galerkin methods. Advance computation of

$$\sigma_{jk}\frac{h_{jk}\bar{u}_{jk}}{2}\big(\coth(\beta_{jk})-\sigma_{jk}\big)$$

for each edge improves performance of the method.

II.4. Algebraic interpretation

From Lemma II.1 it follows that in the pure diffusion limit the linear system (29) reduces to the standard stiffness matrix of the weak Poisson equation and

$$\mathbf{K}_{ij} = \int_{\Omega} \nabla N_j \cdot \nabla N_i \, dV; \quad \forall \boldsymbol{v}_i, \boldsymbol{v}_j \in V(\Omega)\backslash V(\Gamma_D)\,.$$

Using the exactness property (13) one can show that this stiffness matrix has the factorization

$$\mathbf{K} = \mathbf{D}^T \mathbf{E} \mathbf{D}\,, \tag{31}$$

where $\mathbf{D}$ is the *node-to-edge* connectivity matrix and $\mathbf{E}$ is the Gramm matrix of the edge element basis:

$$\mathbf{E}_{ij,kl} = \int_{\Omega} \vec{W}_{ij} \cdot \vec{W}_{kl} \, dV\,,$$

see [4] for details. Upon inspection of (30) it is clear that in the general case we have similar factorization

$$\mathbf{K} = \mathbf{F}^T \mathbf{E} \mathbf{D}\,, \tag{32}$$

where $\mathbf{F}$ has the same sparsity pattern as $\mathbf{D}$ but the row elements are

$$\frac{h_{kl}\bar{u}_{kl}}{2}(\coth(\beta_{kl}) - 1) \quad \text{and} \quad -\frac{h_{kl}\bar{u}_{kl}}{2}(\coth(\beta_{kl}) + 1)\,,$$

respectively. The stiffness matrix (31) generalizes a Laplacian stencil on uniform grids to unstructured grids. Therefore, we can interpret (32) as a generalization of an exponentially fitted stencil for the advection-diffusion operator.

III. Computational study

In all numerical examples Ω is the unit square $[0,1]^2$, $\Gamma_D = \partial\Omega$, $\Gamma_N = \emptyset$, and $K_h(\Omega)$ is a conforming, logically Cartesian, but not necessarily uniform, finite element partition of Ω into quadrilateral elements K_s. For each example problem we specify the advective velocity $\mathbf{u}$, the Dirichlet boundary data g, and the forcing term f. To increase or decrease the Peclet number we vary the value of the diffusion coefficient ε. The boundary $\Gamma_D = \Gamma_B \cup \Gamma_T \cup \Gamma_L \cup \Gamma_R$, where

$$\Gamma_B = \{(x,y)\,|\,0 \le x \le 1; y = 0\}; \quad \Gamma_T = \{(x,y)\,|\,0 \le x \le 1; y = 1\}$$

are the bottom and top sides of Ω and

$$\Gamma_L = \{(x,y)\,|\,0 \le y \le 1; x = 0\}; \quad \Gamma_R = \{(x,y)\,|\,0 \le y \le 1; x = 1\}$$

are the left and the right sides of Ω, respectively

III.1. Numerical methods

Our numerical study compares the EFF-PG method (26) with SUPG and the classical artificial diffusion (AD) method. For bilinear elements the SUPG formulation does not include a second-order term:

$$\int_\Omega \varepsilon\nabla\phi_h\cdot\nabla\psi_h dV + \int_\Omega (\mathbf{u}\cdot\nabla\phi_h)(\psi_h + \tau\mathbf{u}\cdot\nabla\psi_h)dV = \int_\Omega f(\psi_h + \tau\mathbf{u}\cdot\nabla\psi_h)dV\,. \tag{33}$$

A common choice for the stabilization parameter is [6, 14]

$$\tau_K = \frac{h_K}{2|\mathbf{u}|}\left(\coth Pe_K - \frac{1}{Pe_K}\right) \tag{34}$$

where

$$Pe_K = \frac{|\mathbf{u}|h_K}{2\varepsilon}$$

is the element Peclet number. The artificial diffusion method is [13, p.181]

$$\int_\Omega (\varepsilon+\nu)\nabla\phi_h\cdot\nabla\psi_h dV + \int_\Omega (\mathbf{u}\cdot\nabla\phi_h)\psi_h dV = \int_\Omega f\psi_h dV\,, \qquad \nu = \begin{cases} h & \text{if } \varepsilon < h \\ 0 & \text{otherwise}\end{cases}\,. \tag{35}$$

III.2. Example problems

Example 1.

This example is a *manufactured solution* $\phi = x^3 - y^2$. Substitution of this polynomial into the PDE (1) defines the boundary data and the forcing term.

Example 2.

This example specializes the *double-glazing* test problem [8, Example 3.1.4, p.119] to the unit square:

$$\mathbf{u} = \begin{pmatrix} 2(2y-1)(1-(2x-1)^2) \\ -2(2x-1)(1-(2y-1)^2)\end{pmatrix}; \quad f=0; \quad g = \begin{cases} 1 & \text{on } \Gamma_R \\ 0 & \text{on } \Gamma_B\cup\Gamma_T\cup\Gamma_L\end{cases}\,. \tag{36}$$

Problem (36) models temperature distribution in a cavity with a “hot” external wall (Γ_R). The discontinuities at the two corners of the hot wall create boundary layers near its corners.

Example 3.

This example specializes the *constant advection* test case [8, Example 3.1.3, p.118] to the unit square:

$$\mathbf{u} = \begin{pmatrix} -\sin\pi/6 \\ \cos\pi/6\end{pmatrix}; \quad f=0; \quad g = \begin{cases} 0 & \text{on } \Gamma_L\cup\Gamma_T\cup(\Gamma_B\cap\{x\le 0.5\}) \\ 1 & \text{on } \Gamma_R\cup(\Gamma_B\cap\{x>0.5\})\end{cases}\,. \tag{37}$$

Discontinuity in the boundary data leads to an internal layer of width $O(\sqrt{\varepsilon})$. Near Γ_T the solution of (37) develops an exponential boundary layer to match the prescribed boundary data on Γ_T.

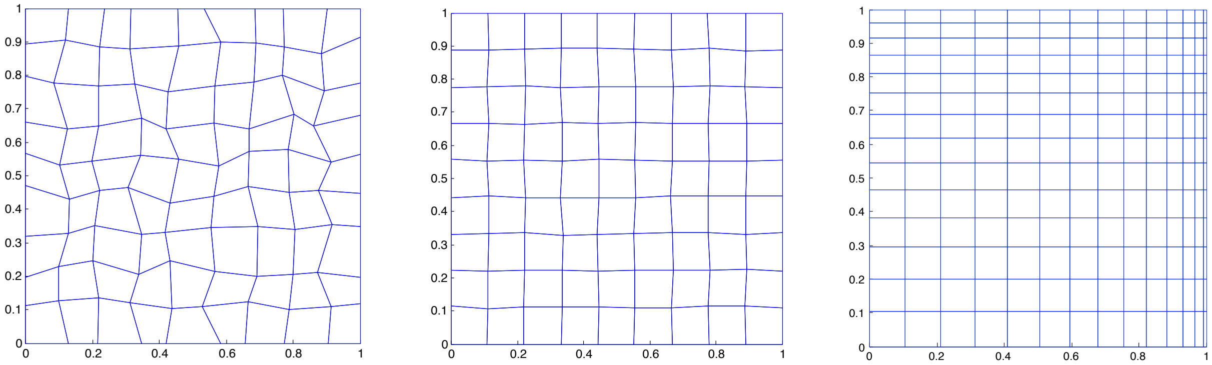


Figure 1. The non-uniform grids for the computational studies. Left pane: $O(1)$ "rough" grid. Center pane: $O(h)$ "smooth" grid. Right pane: "tensor" grid for $\gamma = 0.9$.

III.3. Computational grids

Our numerical study employs three different types of structured quadrilateral grids. Coordinate maps

$$x_{ij} = x(\xi_i, \eta_j, \gamma), \quad y_{ij} = y(\xi_i, \eta_j, \gamma), \qquad 0 \le i \le M_x, \quad 0 \le j \le M_y \,, \tag{38}$$

specify the mesh node positions. In (38) M_x and M_y are the numbers of cells in x and y direction, respectively, γ is real parameter, and

$$\xi_i = \frac{i}{M_x}, \quad i = 0, \ldots, M_x; \qquad \text{and} \qquad \eta_j = \frac{j}{M_y}, \quad j = 0, \ldots, M_y; \tag{39}$$

are the coordinates of an initial (uniform) grid, respectively.

The first grid type is uniform partition of Ω into squares with coordinate maps

$$x(\xi_i, \eta_j, \gamma) = \xi_i \quad \text{and} \quad y(\xi_i, \eta_j, \gamma) = \eta_j$$

The second one is random perturbation of an initial uniform grid with coordinate maps

$$x(\xi_i, \eta_j, \gamma) = \xi_i + 0.25h(r_x h^\gamma); \quad y(\xi_i, \eta_j, \gamma) = \eta_j + 0.25h(r_y h^\gamma)\,. \tag{40}$$

In (40) r_x, r_y are uniformly distributed random numbers in $[-1, 1]$, and $\gamma \ge 0$ is the strength of the perturbation. The nodes on the vertical sides are not allowed to move horizontally and the nodes on the horizontal sides are not allowed to move vertically. We use two levels of perturbations resulting in "rough" and "smooth" non-uniform grids. The "rough" grids correspond to $\gamma = 0$. This value allows the x and y coordinates of the mesh nodes to move up to 1/4 of the initial uniform element size along their respective coordinate axes. The smooth grids correspond to $\gamma = 1$. In this case the coordinate movement is limited to h times 1/4 of the element size. We also refer to the "rough" and "smooth" grids as $O(1)$ and $O(h)$ perturbations of the initial uniform grid; see Fig. 1.

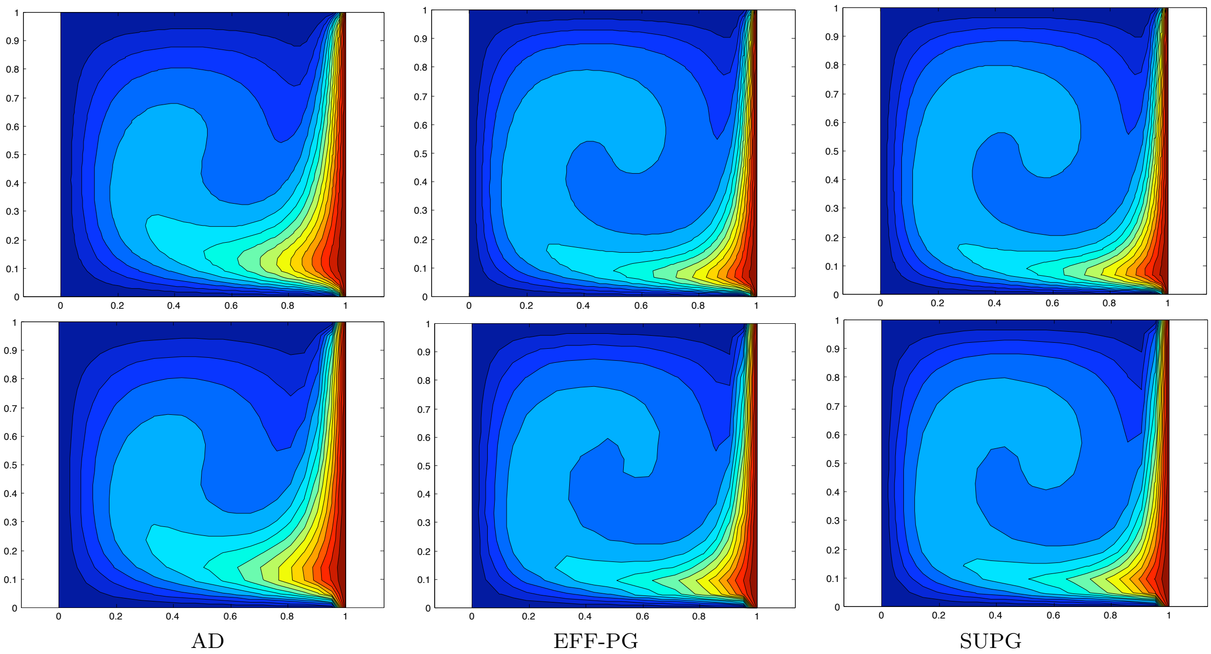


Figure 2. Solution of the double glazing advection test problem (36) with $\varepsilon = 0.01$ by artificial diffusion (left); EFF-PG (middle) and SUPG (right) on 33×33 randomly perturbed grids. Top row: $O(h)$ grid. Bottom row: $O(1)$ grid.

The third grid type has coordinate maps

$$x(\xi,\eta,\gamma) = (1-\alpha(\gamma))\xi + \alpha(\gamma)\xi^3; \quad y(\xi,\eta,\gamma) = (1-\alpha(\gamma))\eta + \alpha(\gamma)\eta^2; \quad \alpha(\gamma) = \frac{\sin(4\pi\gamma)}{2}\,, \tag{41}$$

where $0 \le \gamma \le 1$. The coordinate maps (41) generate a sequence of rectangular, affine tensor-product grids; see the right pane in Fig. 1.

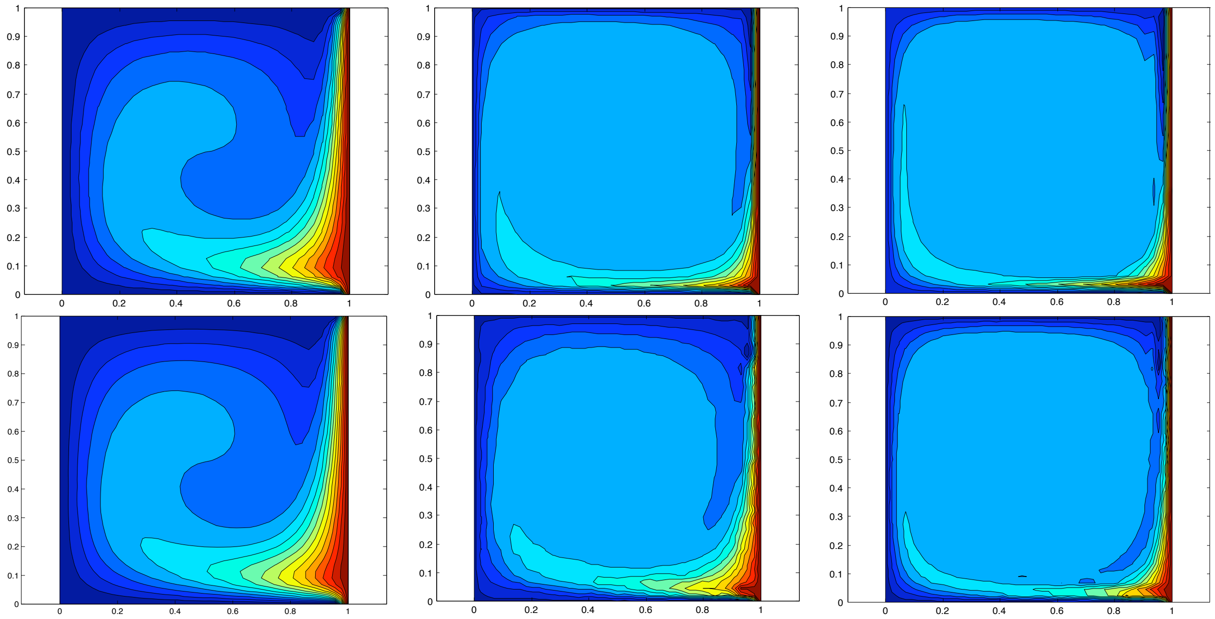
III.4. Numerical results

III.4.1. Robustness.

This study includes of two sets of numerical experiments. In the first set we compare EFF-PG, SUPG and AD solutions of the double glazing and the constant advection problems for four different combinations of grids and ε values. Specifically, we solve problems with "low" ($\varepsilon = 0.01$) and "high" ($\varepsilon = 0.001$) Peclet numbers on"smooth" and "rough" 33×33 grids ($O(h)$ and $O(1)$ random perturbations of an initial uniform grid, respectively).

Example 2: Double glazing.

For $\varepsilon = 0.01$ the largest Peclet numbers on the "smooth" $O(h)$ and "rough" $O(1)$ grids are 1.55 and 1.97, respectively. Figure 2 shows the corresponding EFF-PG, SUPG and AD solutions. We see that in the "low" Peclet number case the EFF-PG and SUPG solutions are very close on both grids. Also, the effects from the additional dissipation in the AD method are visible but the AD solution remains qualitatively close to the EFF-PG and SUPG solutions.



AD EFF-PG SUPG

Figure 3. Solution of the double glazing advection test problem (36) with $\varepsilon = 0.001$ by artificial diffusion (left); EFF-PG (middle) and SUPG (right) on 33×33 randomly perturbed grids. Top row: $O(h)$ grid. Bottom row: $O(1)$ grid.

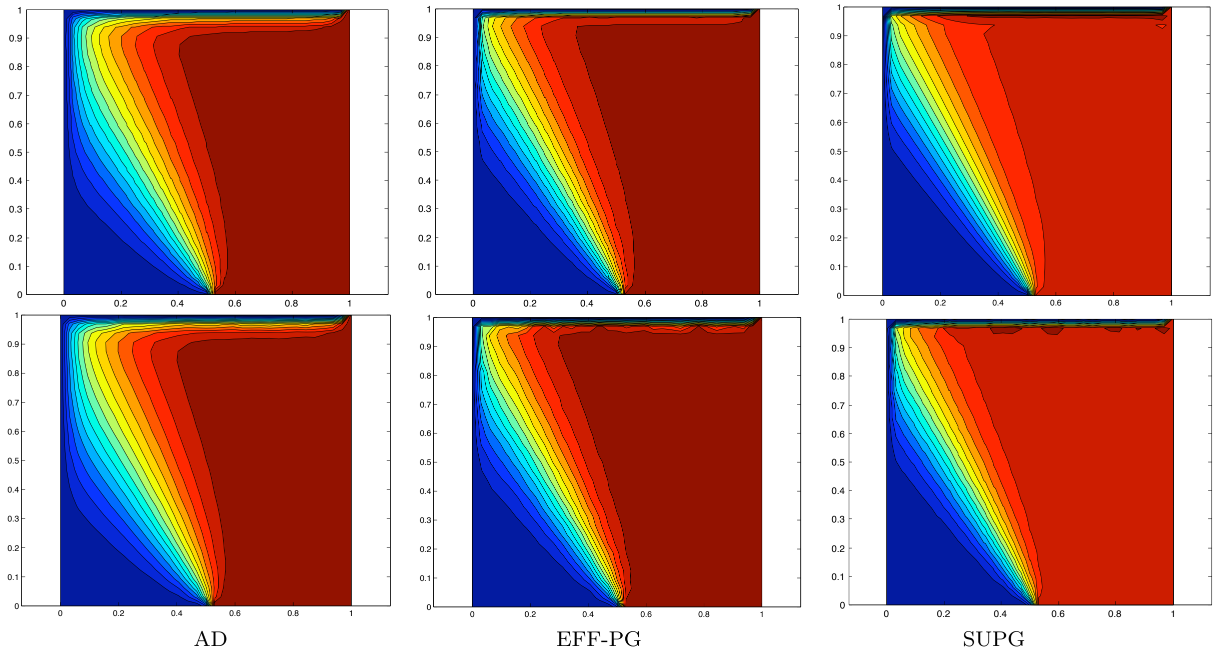
For $\varepsilon = 0.001$ the largest Peclet numbers on the $O(h)$ and $O(1)$ grids scale by 10 to 15.5 and 19.7, respectively. The plots in Figure 3 show that the EFF-PG solution remains qualitatively similar to the SUPG solution and captures the characteristic features of the exact solution. The AD method is clearly more dissipative than the EFF-PG and smears these features.

Example 3: Constant advection.

For $\varepsilon = 0.01$ the largest Peclet numbers on the "rough" and "smooth' grids are 1.07 and 0.79, respectively. In the "low" Peclet number case the EFF-PG, SUPG and AD methods reprise the performance from the double glazing case and produce qualitatively similar solutions; see Figure 4. As before, the AD solution is slightly more dissipative than the EFF-PG and SUPG solutions.

In the "high" Peclet number case $\max Pe = 10.7$ on the "rough" grid and $\max Pe = 7.9$ on the "smooth" grid. Figure 5 confirms that the AD solution remains the most diffusive and significantly smears solution features. The EFF-PG solution is less dissipative and is free of significant spurious oscillations on the "rough" grid.

However, the SUPG solution exhibits spurious oscillations along the internal and the boundary layers. Surface plots in the rightmost column in Figure 6 show that the overshoots in the SUPG solution are significant. This suggests that definition (34) provides insufficient stabilization for this test problem. Examination of this issue is beyond the scope of our paper. We refer to [14] for a more in-depth discussion of reasons to redefine (34). Here, our main points are to illustrate (a) the dependence of the SUPG solution on the definition of τ, and (b) the need



AD EFF-PG SUPG

Figure 4. Solution of the constant advection test problem (37) with $\varepsilon = 0.01$ by artificial diffusion (left); EFF-PG (middle) and SUPG (right) on 33×33 randomly perturbed grids. Top row: $O(h)$ grid. Bottom row: $O(1)$ grid.

to adjust this definition depending on the problem. In particular, the scaled asymptotic limit

$$\tau_K = 2 \lim_{Pe \to \infty} \frac{h_K}{2|\mathbf{u}|} \left(\coth Pe_K - \frac{1}{Pe_K} \right) = \frac{h_K}{|\mathbf{u}|} \tag{42}$$

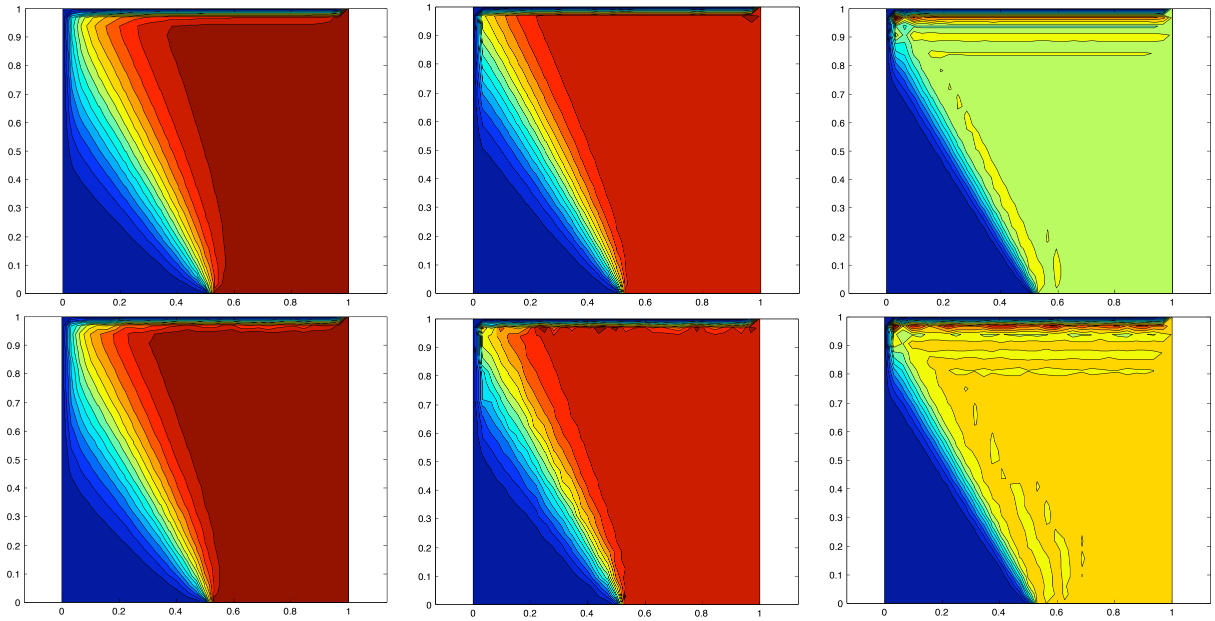
tends to work better for the constant advection test case, but needs to be switched off manually in diffusion-dominated regions; see Figure 6. We note that the EFF-PG method does not require any modifications for the constant advection test.

The second part of the robustness study compares the EFF-PG, SUPG and AD solutions of the double glazing and the constant advection problems for three different values of ε on the "tensor" mesh. The diffusivity values are $\varepsilon = 0.01$, $\varepsilon = 0.001$ and $\varepsilon = 0.0005$. The maximal Peclet numbers for these values are 2.12, 21.25 and 42.51, respectively. Figure 7 presents the results of this study. As in the first study, the AD solution is very diffusive and does not capture well the characteristic features of the solution.

The EFF-PG solution and the SUPG solution are nearly undistinguishable for $\varepsilon = 0.01$. However, at $\varepsilon = 0.001$ we see an onset of spurious oscillations in the SUPG solution, which propagate throughout the computational domain as ε is further decreased. In contrast, the EFF-PG solution remains free of spurious oscillations and resolves the solution features.

III.4.2. Accuracy.

The EFF-PG method (24) uses an exponentially fitted numerical flux. Consequently, it is not a residual-based formulation and we cannot expect that its convergence rates will match the best possible rates for bilinear



AD EFF-PG SUPG

Figure 5. Solution of the constant advection test problem (37) with $\varepsilon = 0.001$ by artificial diffusion (left); EFF-PG (middle) and SUPG (right) on 33×33 randomly perturbed grids. Top row: $O(h)$ grid. Bottom row: $O(1)$ grid.

Table 1. Comparison of the L^2-norm and H^1-seminorm convergence rates of the exponentially fitted flux (EFF) method (26) and the classical artificial diffusion.

Method	EFF-PG method (24)		Artificial diffusion	
Grid	L^2 error	H^1 error	L^2 error	H^1 error
32	0.4484969E-02	0.7733365E-01	0.1352677E-01	0.1308387E+00
64	0.2261091E-02	0.5182587E-01	0.7365740E-02	0.9328349E-01
128	0.1072332E-02	0.3429058E-01	0.3858037E-02	0.6498163E-01
Rate	1.076	0.600	0.933	0.522

elements. However, the comparative study shows that (24) is less dissipative than the artificial diffusion method. As a result, the new method should be at least as accurate as artificial diffusion.

To test this conjecture we compare convergence rates of the EFF-PG and the artificial diffusion method. We use the manufactured solution from Example 1, $\varepsilon = 0.0005$ and the constant advective field from Example 3. The data in Table 1 confirms that (26) is first-order accurate. The L^2 errors of the new method also appear to be a factor of 3 less than the artificial diffusion error.

IV. Conclusions

One can interpret the EFF-PG method as a marriage of the Scharfetter-Gummel idea [18] with the special properties of exact sequences of finite element spaces. From the former, EFF-PG borrows the idea of simplified advection-diffusion equations on the edges to produce exponentially fitted edge fluxes. The latter motivate the use of edge elements to expand the edge fluxes into an $H(curl)$-conforming flux field.

The combination of these two ideas sets the EFF-PG apart from the existing approaches and results in a parameter-free stabilized method. We demonstrate that EFF-PG performs well for a wide range of problems and grids without any user intervention or calibration. The method is first-order accurate but less dissipative than the classical artificial diffusion. Extension of the EFF-PG method to higher-order edge elements is in progress and will be reported in a forthcoming paper.

ACKNOWLEDGEMENTS

The authors would like to thank the Advanced Scientific Computing Research program of the DoE Office of Science and the ASC program of the NNSA for support of this research. We also benefitted from numerous discussions with out colleagues S. Gao, G. Hennigan, L. Musso, and T. Smith.

References

[1] Lutz Angermann and Song Wang. Three-dimensional exponentially fitted conforming tetrahedral finite elements for the semiconductor continuity equations. *Applied Numerical Mathematics*, 46(1):19 – 43, 2003.

[2] D. N. Arnold, R. S. Falk, and R. Winther. Finite element exterior calculus, homological techniques, and applications. *Acta Numerica*, 15:1–155, 2006.

[3] Santiago Badia and Ramon Codina. Analysis of a stabilized finite element approximation of the transient convection-diffusion equation using an ale framework. *SIAM Journal on Numerical Analysis*, 44(5):2159–2197, 2006.

[4] P. Bochev and M. Hyman. Principles of mimetic discretizations. In D. N. Arnold, P. Bochev, R. Lehoucq, R. Nicolaides, and M. Shashkov, editors, *Compatible Discretizations, Proceedings of IMA Hot Topics Workshop on Compatible Discretizations*, volume IMA 142, pages 89–120. Springer Verlag, 2006.

[5] Franco Brezzi, Marie-Odile Bristeau, Leopoldo P. Franca, Michel Mallet, and Gilbert RogÈ. A relationship between stabilized finite element methods and the Galerkin method with bubble functions. *Computer Methods in Applied Mechanics and Engineering*, 96(1):117 – 129, 1992.

[6] Alexander N. Brooks and Thomas J.R. Hughes. Streamline upwind/petrov-galerkin formulations for convection dominated flows with particular emphasis on the incompressible navier-stokes equations. *Computer Methods in Applied Mechanics and Engineering*, 32(1Äì3):199 – 259, 1982.

[7] Ramon Codina. Comparison of some finite element methods for solving the diffusion-convection-reaction equation. *Computer Methods in Applied Mechanics and Engineering*, 156(1-4):185 – 210, 1998.

[8] H. C. Elman, D. J. Silvester, and A. J. Wathen. *Finite Elements and Fast Iterative Solvers with Applications in Incompressible Fluid Dynamics*. Numerical Mathematics and Scientific Computation. Oxford University

Press, 2005.

[9] L. P. Franca and A. Russo. Recovering SUPG using Petrov-Galerkin formulations enriched with adjoint residual free bubbles. *Comput. Meth. Appl. Mech. Eng.*, 182:333–339, 2000.

[10] I. Harari and T. J. R. Hughes. What are C and h?: Inequalities for the analysis and design of finite element methods. *Comput. Meth. Appl. Mech. Eng.*, 97:157–192, 1992.

[11] T. J. R. Hughes and A. Brooks. A theoretical framework for Petrov-Galerkin methods with discontinuous weighting functions: Application to the streamline-upwind procedure. In R. H. Gallagher et al, editor, *Finite Elements in Fluids*, volume 4, pages 47–65, New York, 1982. J. Wiley & Sons.

[12] Thomas J.R. Hughes, Michel Mallet, and Mizukami Akira. A new finite element formulation for computational fluid dynamics: Ii. beyond supg. *Computer Methods in Applied Mechanics and Engineering*, 54(3):341 – 355, 1986.

[13] C. Johnson. *Numerical Solution of Partial Differential Equations by the Finite Element Method.* Cambridge University Press, 1992.

[14] P. Knobloch. On the definition of the SUPG parameter. *ETNA*, 32:76–89, 2008.

[15] M. J. Martinez. Comparison of galerkin and control volume finite element for advection-diffusion problems. *International Journal for Numerical Methods in Fluids*, 50(3):347–376, 2006.

[16] J. C. Nedelec. Mixed finite elements in r^3. *Numerische Mathematik*, 35:315–341, 1980. 10.1007/BF01396415.

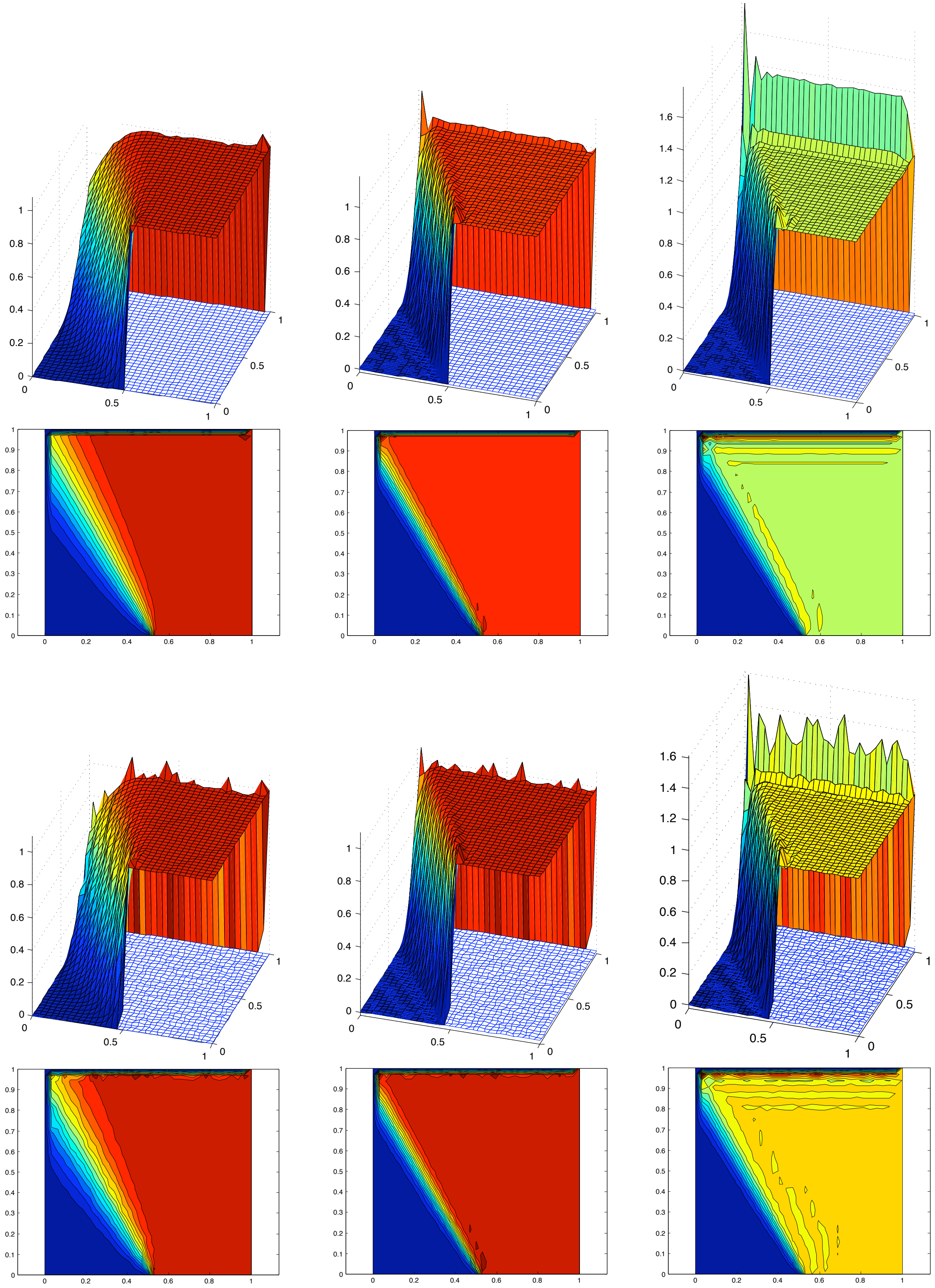
[17] Riccardo Sacco. Exponentially fitted shape functions for advection-dominated flow problems in two dimensions. *Journal of Computational and Applied Mathematics*, 67(1):161 – 165, 1996.

[18] D.L. Scharfetter and H.K. Gummel. Large-signal analysis of a silicon read diode oscillator. *Electron Devices, IEEE Transactions on*, 16(1):64 – 77, jan 1969.

[19] D. Z. Turner, K. B. Nakshatrala, and K. D. Hjelmstad. A stabilized formulation for the advection-diffusion equation using the generalized finite element method. *International Journal for Numerical Methods in Fluids*, 66(1):64–81, 2011.

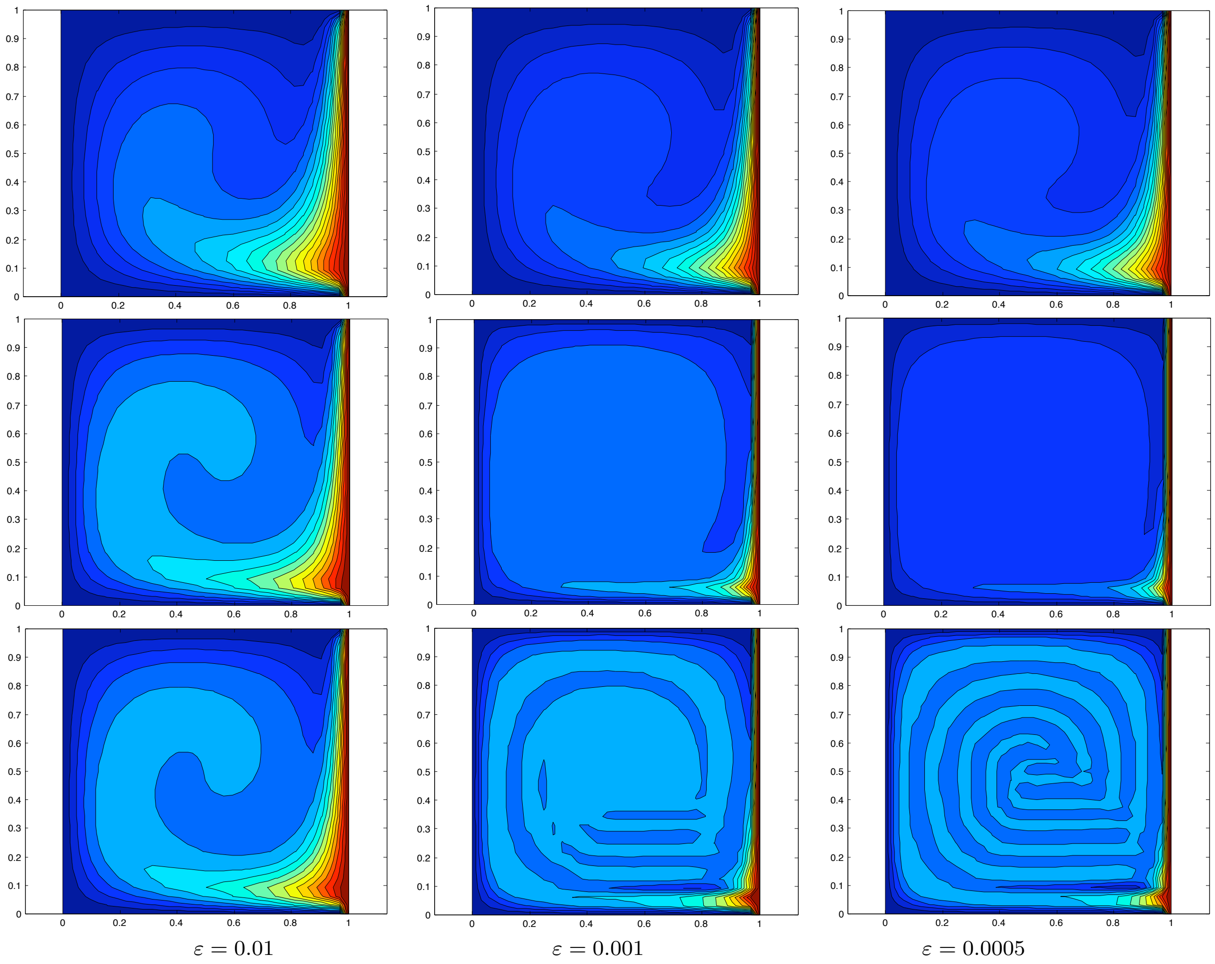
[20] Song Wang. A novel exponentially fitted triangular finite element method for an advection-diffusion problem with boundary layers. *Journal of Computational Physics*, 134(2):253 – 260, 1997.

[21] Song Wang. A new exponentially fitted triangular finite element method for the continuity equations in the drift-diffusion model of semiconductor devices. *M2AN*, 33(1):99–112, 1999.



EFF-PG SUPG-2 SUPG-1

Figure 6. Solution of the constant advection test problem (37) with $\varepsilon = 0.001$ by EFF-PG (left), SUPG with (42) (center) and SUPG with (34) (right) on 33×33 randomly perturbed grids. Top two rows: $O(h)$ grid. Bottom two rows: $O(1)$ grid.



$\varepsilon = 0.01$ $\varepsilon = 0.001$ $\varepsilon = 0.0005$

Figure 7. Solution of the double glazing advection test problem (36) on 33×33 tensor product grid (41) with $\gamma = 0.9$ and decreasing values of ε. Top row: AD solution. Middle row: EFF-PG solution. Bottom row: SUPG solution.